Sentiment Analysis with Python: Bag of Words

2025年7月19日 9:18

1.数据集：IMDB movie review dataset for sentiment analysis

50000个数据，平均分为积极和消极两部分

2.数据处理：

1）Data Cleaning:

仅考虑词汇，去除特殊符号、数字等

大小写归一化

2）Stop Words Removal:

去除对于句子没有意义的词语

3）Stemming

将单词还原为原始形式

40K作训练集 10K作测试集

3.Bag of Words features (BOW)：

通过考虑词汇的出现将文本转换问数字形式

考虑1）词汇表 2）给定文本中单词的出现频率

忽略词语出现的顺序以及语法

通过sklearn库中的CountVectorize实现

仅通过训练集来构建词汇表，且同一词汇表将被应用于测试集

4.逻辑回归：

1）Unigram bag-of-words features:

仅考虑单个词汇

	precision	recall	f1-score	support
Negative	0.88	0.88	0.88	4993
Positive	0.88	0.88	0.88	5007

	precision	recall	f1-score	support
Negative	0.88	0.88	0.88	4993
Positive	0.88	0.88	0.88	5007
accuracy			0.88	10000
macro avg	0.88	0.88	0.88	10000
weighted avg	0.88	0.88	0.88	10000

```
[[4395  598]
 [ 585 4422]]
```

2）Unigrams + Bigrams

	precision	recall	f1-score	support
Negative	0.90	0.89	0.90	4993
Positive	0.89	0.90	0.90	5007
accuracy			0.90	10000
macro avg	0.90	0.90	0.90	10000
weighted avg	0.90	0.90	0.90	10000

准确率略有提升

3）Unigrams + Bigrams + Trigrams

	precision	recall	f1-score	support
Negative	0.90	0.89	0.89	4993
Positive	0.89	0.90	0.90	5007
accuracy			0.90	10000
macro avg	0.90	0.90	0.90	10000
weighted avg	0.90	0.90	0.90	10000

无明显提升

5.线性支持向量机（LSVM）

1）Unigrams

	precision	recall	f1-score	support
Negative	0.86	0.86	0.86	4993
Positive	0.86	0.86	0.86	5007
accuracy			0.86	10000
macro avg	0.86	0.86	0.86	10000
weighted avg	0.86	0.86	0.86	10000

2）Unigrams + Bigrams

	precision	recall	f1-score	support
Negative	0.90	0.90	0.90	4993
Positive	0.90	0.90	0.90	5007
accuracy			0.90	10000
macro avg	0.90	0.90	0.90	10000
weighted avg	0.90	0.90	0.90	10000

3）Unigrams + Bigrams + Trigrams

	precision	recall	f1-score	support
Negative	0.90	0.89	0.89	4993
Positive	0.89	0.90	0.90	5007
accuracy			0.89	10000
macro avg	0.89	0.89	0.89	10000
weighted avg	0.89	0.89	0.89	10000

6.朴素贝叶斯（NB）

1)Unigrams

	precision	recall	f1-score	support
Negative	0.84	0.87	0.86	4993
Positive	0.87	0.83	0.85	5007
accuracy			0.85	10000
macro avg	0.85	0.85	0.85	10000
weighted avg	0.85	0.85	0.85	10000

2)Unigrams + Bigrams

	precision	recall	f1-score	support
Negative	0.87	0.89	0.88	4993
Positive	0.89	0.87	0.88	5007
accuracy			0.88	10000
macro avg	0.88	0.88	0.88	10000
weighted avg	0.88	0.88	0.88	10000

3)Unigrams + Bigrams + Trigrams

	precision	recall	f1-score	support
Negative	0.88	0.89	0.89	4993
Positive	0.89	0.88	0.88	5007
accuracy			0.89	10000
macro avg	0.89	0.89	0.89	10000
weighted avg	0.89	0.89	0.89	10000

Sentiment Analysis with Python : TFIDF features

2025年7月20日 10:45

1.TFIDF：term-frequency-Inverse-document-frequency

1)Term Frequency：给定词语在给定文本中的出现次数

仅使用term-frequency的问题：一些不相关的词语出现次数多

2）IDF:

考虑一个词语在所有文本中的出现情况，如果只在一些文本中出现，则获得较高IDF值，如果在大部分文本中出现，则获得较低IDF值

不考虑词语在某一个文本中出现的频率

3）TFIDF:

根据TF和IDF对文本进行评估。

TFIDF分数说明一个词语t在一个文本d中的重要性（很多文本D同时存在）

tfidf(t,d,D) = tf(t,d) * idf(t,D)

2.Logistic Regression

1)Unigrams:

```
              precision    recall  f1-score   support

    Negative       0.89      0.88      0.89      4993
    Positive       0.88      0.90      0.89      5007

    accuracy                           0.89     10000
   macro avg       0.89      0.89      0.89     10000
weighted avg       0.89      0.89      0.89     10000

[[4397  596]
 [ 519 4488]]
```

2)Unigrams + Bigrams

```
              precision    recall  f1-score   support

    Negative       0.90      0.88      0.89      4993
    Positive       0.88      0.90      0.89      5007

    accuracy                           0.89     10000
   macro avg       0.89      0.89      0.89     10000
weighted avg       0.89      0.89      0.89     10000

[[4393  600]
 [ 512 4495]]
```

3)Unigrams + Bigrams + Trigrams

```
              precision    recall  f1-score   support

    Negative       0.89      0.88      0.88      4993
    Positive       0.88      0.89      0.89      5007

    accuracy                           0.88     10000
   macro avg       0.88      0.88      0.88     10000
weighted avg       0.88      0.88      0.88     10000

[[4399  594]
 [ 558 4449]]
```

3.LSVM

1)Unigrams

```
              precision    recall  f1-score   support

    Negative       0.90      0.89      0.89      4993
    Positive       0.89      0.90      0.89      5007

    accuracy                           0.89     10000
   macro avg       0.89      0.89      0.89     10000
weighted avg       0.89      0.89      0.89     10000

[[4426  567]
 [ 514 4493]]
```

2)Unigrams + Bigrams

```
              precision    recall  f1-score   support

    Negative       0.91      0.89      0.90      4993
    Positive       0.90      0.91      0.90      5007

    accuracy                           0.90     10000
   macro avg       0.90      0.90      0.90     10000
weighted avg       0.90      0.90      0.90     10000

[[4462  531]
 [ 453 4554]]
```

3)Unigrams + Bigrams + Trigrams

```
              precision    recall  f1-score   support

    Negative       0.91      0.89      0.90      4993
    Positive       0.89      0.91      0.90      5007

    accuracy                           0.90     10000
   macro avg       0.90      0.90      0.90     10000
weighted avg       0.90      0.90      0.90     10000

[[4444  549]
 [ 465 4542]]
```

4.MNB

1)Unigrams

```
              precision    recall  f1-score   support

    Negative       0.85      0.88      0.86      4993
    Positive       0.87      0.84      0.86      5007

    accuracy                           0.86     10000
   macro avg       0.86      0.86      0.86     10000
weighted avg       0.86      0.86      0.86     10000

[[4387  606]
 [ 783 4224]]
```

2)Unigrams + Bigrams

```
              precision    recall  f1-score   support

    Negative       0.88      0.90      0.89      4993
    Positive       0.89      0.87      0.88      5007

    accuracy                           0.89     10000
   macro avg       0.89      0.89      0.89     10000
weighted avg       0.89      0.89      0.89     10000

[[4479  514]
 [ 634 4373]]
```

3)Unigrams + Bigrams + Trigrams

```
              precision    recall  f1-score   support

    Negative       0.88      0.89      0.89      4993
    Positive       0.89      0.88      0.89      5007

    accuracy                           0.89     10000
   macro avg       0.89      0.89      0.89     10000
weighted avg       0.89      0.89      0.89     10000

[[4463  530]
 [ 605 4402]]
```

Sentiment Classification with Deep Learning: RNN, LSTM, and CNN

2025年7月21日 10:56

1.数据集准备

1）数据清理

2）词汇表创建

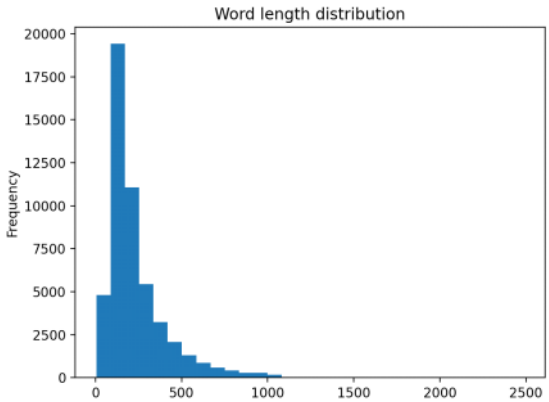
每一个词语对应独特的索引值，将评论文本转换为一串整数

大约有92K的单词，其中前10K的单词就可以覆盖约95%的文章中的单词

3）将评论转换为整数列表

仅使用前10K的单词

将单一评论长度限制为500词，短于500词用特殊索引填充，长于500词仅考虑前500词



根据该图表，500词已经包含大部分评论，随后可考虑延长为1000词

4）输出结果转换为数字形式

2.多层感知机MLP：

1）嵌入层：将数据中的每个单词对应为一个嵌入向量，size=Embedding+size

该实验选择为32

2）对于一条评论，得到一个500*32的数据块，随后扁平化成一个16000的单维向量，用于输入此后的全连接层

3）最后的输出层使用sigmoid激活函数，输出一个0~1之间的概率值，表示该评论为积极的可能性

4)损失函数使用二元交叉熵

5）模型信息

Model: "functional"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 500)	0
embedding (Embedding)	(None, 500, 32)	320,064
flatten (Flatten)	(None, 16000)	0
dense (Dense)	(None, 16)	256,016
activation (Activation)	(None, 16)	0
dropout (Dropout)	(None, 16)	0
dense_1 (Dense)	(None, 8)	136
activation_1 (Activation)	(None, 8)	0
dropout_1 (Dropout)	(None, 8)	0
dense_2 (Dense)	(None, 4)	36
activation_2 (Activation)	(None, 4)	0
dropout_2 (Dropout)	(None, 4)	0
dense_3 (Dense)	(None, 1)	5
activation_3 (Activation)	(None, 1)	0

Total params: 576,257 (2.20 MB)
Trainable params: 576,257 (2.20 MB)
Non-trainable params: 0 (0.00 B)

6）训练过程以及结果：

```
Epoch 1/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 3s 15ms/step - accuracy: 0.5061 - loss: 0.6947 - val_accuracy: 0.4993 - val_loss: 0.6931
Epoch 2/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 3s 16ms/step - accuracy: 0.5054 - loss: 0.6928 - val_accuracy: 0.5507 - val_loss: 0.6915
Epoch 3/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 2s 15ms/step - accuracy: 0.5209 - loss: 0.6902 - val_accuracy: 0.6233 - val_loss: 0.6711
Epoch 4/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 2s 14ms/step - accuracy: 0.5953 - loss: 0.6602 - val_accuracy: 0.7244 - val_loss: 0.6156
Epoch 5/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 2s 14ms/step - accuracy: 0.6937 - loss: 0.5984 - val_accuracy: 0.8144 - val_loss: 0.5020
Epoch 6/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 2s 15ms/step - accuracy: 0.7851 - loss: 0.5087 - val_accuracy: 0.8589 - val_loss: 0.4356
Epoch 7/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 2s 15ms/step - accuracy: 0.8185 - loss: 0.4568 - val_accuracy: 0.8699 - val_loss: 0.4087
Epoch 8/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 2s 14ms/step - accuracy: 0.8507 - loss: 0.4042 - val_accuracy: 0.8575 - val_loss: 0.4084
Epoch 9/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 2s 15ms/step - accuracy: 0.8683 - loss: 0.3620 - val_accuracy: 0.8546 - val_loss: 0.4103
Epoch 10/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 2s 15ms/step - accuracy: 0.8752 - loss: 0.3413 - val_accuracy: 0.8702 - val_loss: 0.4382
```

3.循环神经网络RNN

1）模型信息

Model: "functional"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 500)	0
embedding (Embedding)	(None, 500, 32)	320,064
bidirectional (Bidirectional)	(None, 100)	8,300
dense (Dense)	(None, 1)	101
activation (Activation)	(None, 1)	0

Total params: 328,465 (1.25 MB)
Trainable params: 328,465 (1.25 MB)
Non-trainable params: 0 (0.00 B)

2）训练过程以及结果

```
Epoch 1/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 21s 124ms/step - accuracy: 0.5219 - loss: 0.6904 - val_accuracy: 0.6409 - val_loss: 0.6410
Epoch 2/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 20s 130ms/step - accuracy: 0.7502 - loss: 0.5228 - val_accuracy: 0.6905 - val_loss: 0.5847
Epoch 3/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 24s 156ms/step - accuracy: 0.8177 - loss: 0.4174 - val_accuracy: 0.8407 - val_loss: 0.3918
Epoch 4/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 22s 141ms/step - accuracy: 0.8787 - loss: 0.2991 - val_accuracy: 0.8259 - val_loss: 0.4289
Epoch 5/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 23s 144ms/step - accuracy: 0.9125 - loss: 0.2346 - val_accuracy: 0.8373 - val_loss: 0.4248
Epoch 6/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 26s 166ms/step - accuracy: 0.9458 - loss: 0.1611 - val_accuracy: 0.8158 - val_loss: 0.4825
Epoch 7/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 29s 187ms/step - accuracy: 0.9684 - loss: 0.1006 - val_accuracy: 0.8220 - val_loss: 0.5327
Epoch 8/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 32s 201ms/step - accuracy: 0.9841 - loss: 0.0573 - val_accuracy: 0.8274 - val_loss: 0.5952
Epoch 9/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 30s 193ms/step - accuracy: 0.9916 - loss: 0.0343 - val_accuracy: 0.8127 - val_loss: 0.6616
Epoch 10/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 33s 209ms/step - accuracy: 0.9861 - loss: 0.0471 - val_accuracy: 0.8251 - val_loss: 0.6961
```

4.LSTM

1）模型信息

Model: "functional"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 500)	0
embedding_1 (Embedding)	(None, 500, 32)	320,064
bidirectional (Bidirectional)	(None, 100)	33,200
dense (Dense)	(None, 1)	101
activation (Activation)	(None, 1)	0

Total params: 353,365 (1.35 MB)
Trainable params: 353,365 (1.35 MB)
Non-trainable params: 0 (0.00 B)

2）训练过程以及结果

```
Epoch 1/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 199s 1s/step - accuracy: 0.6417 - loss: 0.6126 - val_accuracy: 0.8683 - val_loss: 0.3372
Epoch 2/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 285s 2s/step - accuracy: 0.8798 - loss: 0.3062 - val_accuracy: 0.8796 - val_loss: 0.3011
Epoch 3/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 270s 2s/step - accuracy: 0.9106 - loss: 0.2422 - val_accuracy: 0.8892 - val_loss: 0.2884
Epoch 4/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 231s 1s/step - accuracy: 0.9277 - loss: 0.2050 - val_accuracy: 0.8743 - val_loss: 0.3392
Epoch 5/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 307s 2s/step - accuracy: 0.9381 - loss: 0.1790 - val_accuracy: 0.8883 - val_loss: 0.3083
Epoch 6/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 208s 1s/step - accuracy: 0.9496 - loss: 0.1479 - val_accuracy: 0.8857 - val_loss: 0.3154
Epoch 7/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 214s 1s/step - accuracy: 0.9571 - loss: 0.1293 - val_accuracy: 0.8848 - val_loss: 0.3610
Epoch 8/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 240s 2s/step - accuracy: 0.9557 - loss: 0.1307 - val_accuracy: 0.8826 - val_loss: 0.3528
Epoch 9/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 214s 1s/step - accuracy: 0.8670 - loss: 0.2977 - val_accuracy: 0.8777 - val_loss: 0.3613
Epoch 10/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 234s 1s/step - accuracy: 0.9446 - loss: 0.1531 - val_accuracy: 0.8833 - val_loss: 0.3404
```

5. 1D CNN

1）模型信息

Model: "functional"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 500)	0
embedding (Embedding)	(None, 500, 32)	320,064
conv1d (Conv1D)	(None, 498, 50)	4,850
max_pooling1d (MaxPooling1D)	(None, 249, 50)	0
conv1d_1 (Conv1D)	(None, 247, 40)	6,040
max_pooling1d_1 (MaxPooling1D)	(None, 123, 40)	0
conv1d_2 (Conv1D)	(None, 121, 30)	3,630
max_pooling1d_2 (MaxPooling1D)	(None, 60, 30)	0
conv1d_3 (Conv1D)	(None, 58, 30)	2,730
max_pooling1d_3 (MaxPooling1D)	(None, 29, 30)	0
flatten (Flatten)	(None, 870)	0
dense (Dense)	(None, 20)	17,420
dropout (Dropout)	(None, 20)	0
dense_1 (Dense)	(None, 1)	21
activation (Activation)	(None, 1)	0

Total params: 354,755 (1.35 MB)
Trainable params: 354,755 (1.35 MB)
Non-trainable params: 0 (0.00 B)

2）训练过程以及结果

```
Epoch 1/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 7s 36ms/step - accuracy: 0.5551 - loss: 0.6528 - val_accuracy: 0.8842 - val_loss: 0.2931
Epoch 2/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 6s 35ms/step - accuracy: 0.8997 - loss: 0.2615 - val_accuracy: 0.8951 - val_loss: 0.2645
Epoch 3/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 5s 35ms/step - accuracy: 0.9351 - loss: 0.1837 - val_accuracy: 0.8891 - val_loss: 0.2976
Epoch 4/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 6s 35ms/step - accuracy: 0.9533 - loss: 0.1389 - val_accuracy: 0.8971 - val_loss: 0.2905
Epoch 5/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 6s 35ms/step - accuracy: 0.9758 - loss: 0.0821 - val_accuracy: 0.8944 - val_loss: 0.3560
Epoch 6/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 6s 35ms/step - accuracy: 0.9858 - loss: 0.0505 - val_accuracy: 0.8890 - val_loss: 0.4660
Epoch 7/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 5s 34ms/step - accuracy: 0.9913 - loss: 0.0308 - val_accuracy: 0.8907 - val_loss: 0.5726
Epoch 8/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 6s 39ms/step - accuracy: 0.9954 - loss: 0.0186 - val_accuracy: 0.8870 - val_loss: 0.6894
Epoch 9/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 6s 40ms/step - accuracy: 0.9966 - loss: 0.0131 - val_accuracy: 0.8888 - val_loss: 0.7195
Epoch 10/10
157/157 ━━━━━━━━━━━━━━━━━━━━ 6s 40ms/step - accuracy: 0.9930 - loss: 0.0224 - val_accuracy: 0.8875 - val_loss: 0.6550
```